キャッシュを用いたパケット転送処理の省電力化および高速化に関する研究

High-Throughput and Low-Power Packet Forwarding using Cache

研究代表者　電気通信大学大学院情報理工学研究科　助教　八巻隼人
Hayato Yamaki

To increase the throughput and power efficiency of core routers is an important issue for future internet because of the explosive increase in the internet traffic amount. To solve this problem, we have proposed packet processing cache (PPC). PPC improves TCAM (ternary content addressable memory) lookups, which is known as the major throughput bottleneck and one of the causes of consuming power in routers. PPC stores the TCAM lookup results into a cache and reduces the number of TCAM accesses by reusing the cached results. The performance of PPC depends on the number of cache misses (it means the situation that the TCAM lookup results are not in the cache) because the cache miss requires the TCAM accesses. For this reason, this research introduces several cache control methods to reduce the number of cache misses. We showed that Elevator Cache (ELC), introduced in this paper as one of our proposed methods, can reduce the number of cache misses by 11.1% on average compared to Least Recently Used (LRU), used in conventional PPCs.

要旨

通信量の爆発的増加が予想される今後のインターネットにおいて，ルータのパケット処理速度の向上および消費電力の削減は重要な課題となっている．この課題の解決に向けて，我々はルータにおけるパケット処理キャッシュ（Packet Processing Cache: PPC）を提案してきた．PPCは，従来の低速かつ高消費電力なTCAM（Ternary Content Addressable Memory）に対するテーブル検索の結果を，低容量だが高速かつ省電力なキャッシュメモリに保存し，再利用することで，パケット処理を高速化，省電力化する．PPCを用いたパケット処理では，いかにTCAMアクセスをキャッシュメモリが代替できるかが性能を大きく左右する．そこで本研究では，PPCにおいてキャッシュメモリに格納するテーブル検索結果を適切に制御することでTCAMアクセス数を削減する，様々な方法を提案した．本報告で紹介するELCはPPCに適用することでキャッシュミスを従来の11.7%削減可能であることを示した．

1．緒言

近年，世界的な電力消費量の増加は地球温暖化の進行を招いており，様々な分野における省電力化が重要な課題となっている[1]．中でもネットワーク機器の消費電力は，現状でも世界総発電量の数%程度を占めているにもかかわらず，今後の情報爆発に伴い増大することが懸念されており，早急な省電力化が求められている[2]．環境省の報告によると，2006年から2025年にかけてインターネット通信量は190倍に，またネットワーク機器消費電力は10倍以上に増大することが予想されている[3]．特にバックボーンネットワークに配置されるコアルータでは，トラフィックが集中することから，高速なパケット転送をより低電力に実現しなければならない．

コアルータにおいて，パケット処理の速度上のボトルネックとなり，なおかつ高い電力を消費する処理にテーブル検索処理がある[4]．ルータはパケットの処理に要する情報をルーティングテーブル，ACL（Access Control List），QoS（Quality of Service）テーブルといった，各種テーブルに保持しており，1パケット毎にこれら全てのテーブルを検索することでパケットを処理する．近年のコアルータは，テーブルをTernary Content Addressable Memory（TCAM）と呼ばれる高い検索性能を有するメモリに格納することで，検索の高速化を実現している．

TCAMは入力データとTCAM内の全キーを一斉に比較することで，1サイクルでTCAM内データのマッチングを行えるが，同容量の一般的なStatic Random Access Memory（SRAM）と比べても，1検索あたり16倍以上の高い電力を要する[5]．特にパケット処理では，パケット毎に複数回テーブル検索を要することから，TCAMへのアクセス頻度が高い．これにより，ルータ全消費電力の40%程度をTCAMが占めていると報告されている[6][7]．また，TCAMは検索性能面での

不足も懸念される．最短パケット長のパケットが連続してルータに到着する最悪ケースでは，TCAM により達成可能なパケット処理速度は 100Gbps 程度となる．400Gbps から 1Tbps を見据えた今後のネットワークにおいては，テーブル検索速度を大幅に改善しなければならない．そのため，今後は TCAM のみに頼らない新たなテーブル検索手法を模索する必要がある．本研究では，TCAM によるテーブル検索の問題点を改善しうるパケット処理キャッシュに着目し，インターネット通信の特性に応じたキャッシュ制御手法を組み込むことで，更なるパケット処理性能の向上を目指す．

２．パケット処理キャッシュ

テーブル検索を高速化，省電力化する一つの方向性として，小規模だが高速かつ低消費電力なキャッシュメモリを用いたパケット処理キャッシュ（Packet Processing Cache: PPC）が注目されている[(8)(9)(10)]．ルータのパケット処理におけるテーブル検索では，パケットヘッダ内の一つあるいは複数のフィールドを検索キーとして用いており，これらの値が等しいパケット群に対して同一のテーブル検索結果が返される．そこで，PPC では多くのテーブル検索に用いられる 5 タプル値（送信元/宛先 IP アドレス，送信元/宛先ポート番号，プロトコル番号）により，パケットをフローと定義する．フローの初回パケットのテーブル検索結果をキャッシュへと保存し，同一フローの 2 パケット目以降をキャッシュにより処理することで，複数回の TCAM アクセスを要するテーブル検索を一度のキャッシュ参照により処理可能とする．キャッシュメモリは TCAM に比べ，アクセス時の遅延と消費エネルギーが低いため，PPC によりテーブル検索の高スループット化，省電力化が期待できる．PPC は TCAM を用いた既存のテーブル検索アーキテクチャの上に追加的に実装可能であり，キャッシュにテーブル検索結果が存在しない（キャッシュミス）時であっても従来と同等のテーブル検索性能が確保できる．

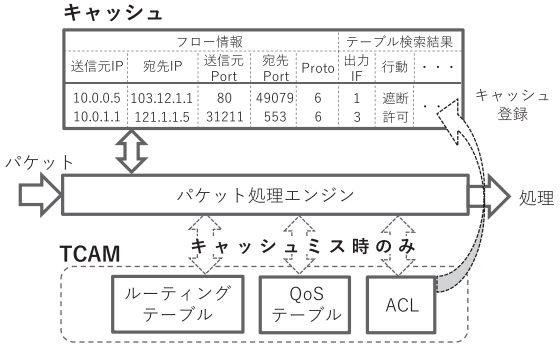

図1 パケット処理キャッシュを用いたテーブル検索処理

パケット処理キャッシュの概要を図 1 に示した．PPC では，前述した 5 タプルによる 13byte のフロー情報をキャッシュタグとして用い，各種テーブルの検索結果をキャッシュデータとして格納する．例えば，パケット処理においてルーティングテーブル，ARP テーブル，ACL，QoS テーブルの検索を要する一般的なルータでは，出力ポート情報として 1byte，宛先 MAC アドレスとして 12byte，フィルタリング結果として 1byte，QoS 情報として 1byte の計 15byte がキャッシュデータとなる．

PPC を用いたパケット処理の消費電力および処理遅延は，メモリの性能とミスの発生に左右される．一般的に，キャッシュは容量を大きくするほどミスの削減を望めるが，容量とアクセス速度，消費電力の間にはトレードオフの関係があるため，必ずしも容量を増やし高いヒット率を得ることが性能向上に繋がるわけではない．近年の汎用プロセッサにおけるキャッシュの容量とメモリアクセス速度は，Intel の公表[(11)]によると L1 キャッシュが 32KB で 1ns 程度，L2 キャッシュが 256KB で 3ns 程度である．一方で，TCAM も 5ns 程度でアクセスできる製品が販売されており，L2 キャッシュと大差ないアクセス速度が得られていることを考慮すると，本機構は L1 規模のメモリへ実装することが望ましい．容量不足によるミスの増大を防ぐためには，キャッシュ空間の効率的な利用が不可欠となる．報告者は，PPC においてキャッシュミスが発生する要因について詳細に分析し[4]，様々なキャッシュ使用効率の向上手法を提案してきた[1][2][3][5]．本報告では，その成果の 1 つとして，キャッシュのライン置き換え方式に着目した研究[5]の内容と成果について報告する．

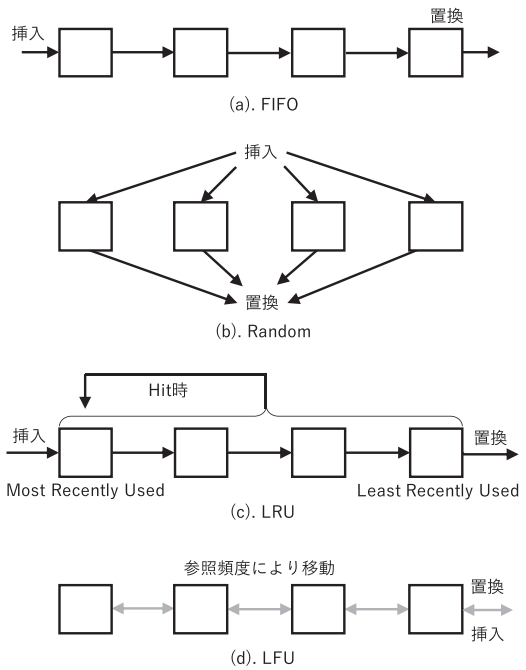

図2 一般的なライン置き換え方式の概要

3．ライン置き換え方式

キャッシュ空間を効率的に利用する上で重要な要素としてライン置き換え方式がある．これは，一つのインデックスに対して複数のエントリが存在するセットアソシアティブキャッシュにおいて，新規エントリを挿入する際に追い出すエントリを決定する方法を表す．以降では，まず多くのキャッシュで採用される一般的な置き換え方式について検討し，PPCに適用した場合の問題点を明らかにする．次に，これらの問題点とフローの特徴から，PPCに適する置き換え方式を検討し，その改善手法を提案する．

3.1　一般的なライン置き換え方式

実装の容易なライン置き換え方式として，単純なキューに用いられるFirst In First Out（FIFO）がある．FIFOでは，最初に入ったデータが最初に取り出される．図2（a）に，1インデックスに4エントリが存在する4wayキャッシュにおけるFIFOの動作を示した．FIFOはキャッシュ内データの再参照が短時間に集中する場合には効果的だが，挿入されてからway数回の新規データ挿入で追い出されるため，新規データ挿入が頻繁に行われる場合やデータの再参照間隔が長い場合には適さない．

Random方式では，ランダムに置換対象エントリが選択される．図2（b）にRandomの概要を示した．本方式はキャッシュのway数によらず簡易なハードウェアで実装できることから，時間的局所性や空間的局所性の弱いデータを扱うキャッシュにおいて用いられることがある．しかしながら，PPCでキャッシュされるフローは強い時間的局所性を持つため，PPCにおいては本方式の効果は小さいと考えられる．

Least Recently Used（LRU）は参照された時間が最も古いエントリを置換対象とする方式である．図2（c）にLRUの概要を示した．FIFOとの違いはデータ再参照の局所性に重きを置いた点である．CPUなど，多くのキャッシュにおいてLRUは高いキャッシュミス削減効果が得られることから，本方式が採用されており，PPCにおいてもこれまで経験的にLRUが用いられてきた．LRUの欠点としては，データが挿入されてから追い出されるまでに最低でもway数回の新規エントリ挿入を経る必要があり，不要なデータ挿入が多いキャッシュにおいてはLRUが最適とは言えない．

Least Frequently Used（LFU）は参照回数の最も少ないエントリを置換対象とする方式である．図2（d）にLFUの概要を示した．LFUはデータの参照回数に焦点を当てた置換方式であり，時間によらず再参照されるデータが多い場合に高い効果を発揮する．しかしながら，PPCにおいては，通信を終えたフローのパケットが再度転送されることはなく，このようなフローのエントリがいつまでも残ることでキャッシュ空間の利用効率が低下することが考えられる．

Optimal Page Replacement Algorithm（OPT）は，後続する全データを知っている理想的な状態で，再参照されないデータまたは最も遅く再参照されるデータを置換対象とする方式である[12]．本方式は，後続するデータを知っているという前提上，実用は難しく，一方で性能は全ての方式の中でも最大となる．OPTの意義は，ライン置き換え方式を検討する上で，理論上の最大性能を把握できることにある．本研究においても，ライン置き換え方式のミス削減効果を可能な限りOPTに近づけることが目的となる．

3.2　最適なライン置き換え方式の検討

図3は，4wayキャッシュにおいて，従来PPCで用いられてきたLRUと理想的な追い出しが可能なOPTそれぞれを適用した場合のキャッシュミス率を測定したシミュレーション結果を示している．本測定に

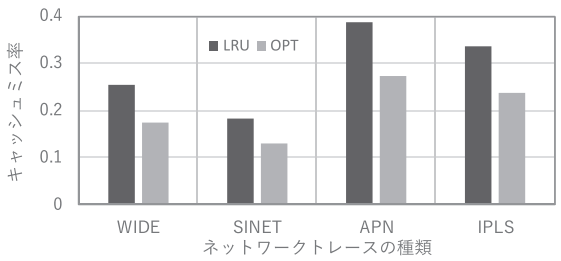

図3 ライン置き換え方式によるミス率の比較

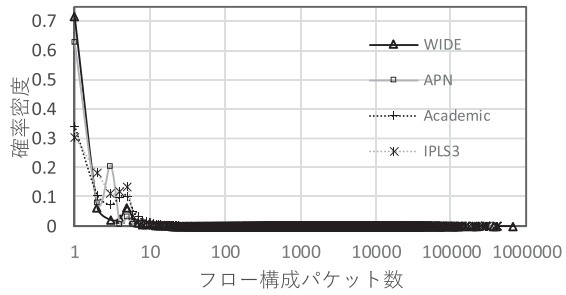

図4 フロー構成パケット数の分布

使用したシミュレータおよびネットワークトレースの詳細は5節で後述する．図3の結果は，従来用いられてきたLRUはPPCにおいて最適とは言えず，OPTのような最適なライン置き換え方式を適用することができれば，トレースによらずまだ30%程度のミス削減を見込める余地があることを示している．そこで，本研究では，パケット処理キャッシュにおいてより適したライン置き換え方式を検討し，上述したミス率の乖離を縮めることを目的とした．

我々は，まず，PPCにおけるキャッシュミスの発生について分析した．キャッシュミスが生じる主な原因は，フローがヒットする前に，他のフローの新規挿入のよって当該フローが追い出されてしまうことである．従って，ヒットする可能性の高いフローを残し，ヒットする可能性の低いフローを優先的に追い出すことで，キャッシュミスの削減が期待できる．ここで，PPCにおけるヒット可能性の高さを考える指標として，フローを構成するパケット数に着目する．多パケットで構成されるフローは多くのキャッシュヒットが期待でき有用であり，一方で，少パケットで構成されるフローは少ないヒット回数しか期待できず有用性が低い．図4は，実ネットワークにおけるフロー構成パケット数の分布を測定した結果である．これによると．トラフィック中の95%

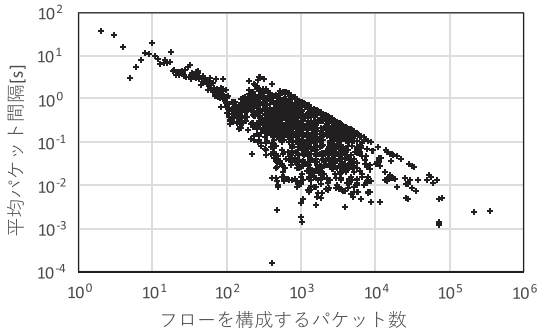

図5 フローにおける構成パケット数と平均パケット間隔の関係

以上のフローが10パケット以下で構成されるショートフローであり，特に1パケットで構成されるフローが全体の半数程度を占めていることがわかる．一方で，10万パケットを超えるロングフローの存在も確認できる．これらのフローのPPCにおけるヒット可能性を考慮すると，ショートフローをキャッシュに残さず，ロングフローを優先的に残すことが，キャッシュ空間を効率的に活用する上でのエントリ制御方針となる．

ここまでの検討により，FIFOやLRUはロングフローに優先度を与える一方でショートフローによりエントリを乱されることが，LFUはショートフローを迅速に追い出す一方で通信の完了したロングフローを残すことが問題だと考えられる．より効果的に置換対象を決定するためには，ショートフローとロングフローの双方を適切に扱う手法を模索する必要がある．

更に，本研究ではフローのパケット間隔に着目した．図5はフローを構成するパケット数と，そのパケット間隔の平均時間との関係を示したグラフである．図5より，フローを構成するパケット数とパケット間隔の間には相関があり，ロングフローのパケット間隔はショートフローの100分の1以下であることがわかる．これは，ショートフローの多くが通信ホスト間でのやりとりであるのに対し，ロングフローは一方的なファイル転送が多いためであると考えられる．従って，ロングフローであるほど最大可能ヒット回数のみならずヒット機会も増える．

4．ライン置き換え方式の提案

本研究ではパケット処理キャッシュに適したライン置き換え方式として2つの方式を検討した．

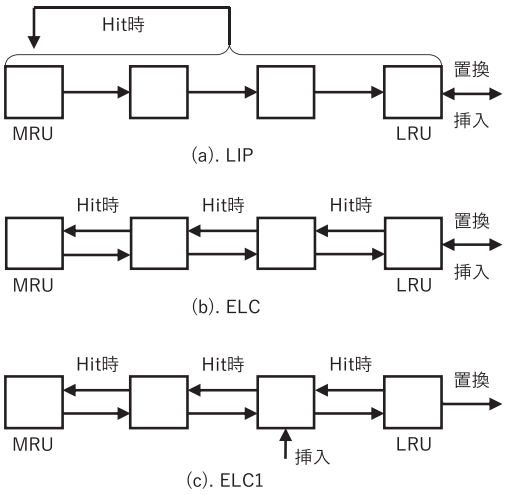

図6 LIP および ELC とその改善手法の概要

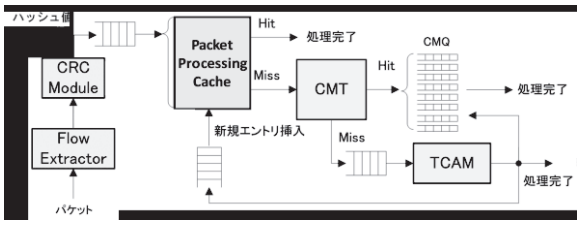

図7 PPC シミュレータのブロック図

表1 シミュレーションで設定したパラメータ

パラメータ	値
キャッシュの総エントリ数	1,024 (=1K)
キャッシュの way 数	4
キャッシュのアクセス遅延	1 ns
TCAM のアクセス遅延	3.3 ns

4.1 LRU Insertion Policy (LIP)

まず,最初に検討する LRU Insertion Policy (LIP) は，Qureshi らにより提案された CPU におけるライン置き換え方式であり，LRU において新規エントリが Most Recently Used (MRU) ポジションに挿入されるのに対し，新規エントリを LRU ポジションに挿入することで，再参照性の低いデータによるキャッシュ汚染を防ぐ[13]. 図6(a)に LIP の概要を示した. 本方式は，LRU と新規エントリの挿入位置が異なるだけで同様のハードウェアにより実装可能であり，再参照性の低いデータが多数存在するプログラムにおいて高い効果を発揮する．しかしながら，再参照性の高いデータであっても追い出されてしまうことから一般的なキャッシュには普及していない．パケット処理キャッシュにおいては，前述したようにロングフローのヒット機会がショートフローに比べ多いため，ロングフローが優先的に残ると期待できる.

4.2 Elevator Cache (ELC)

次に，本論文では Elevator Cache (ELC) を提案する．LIP は再参照されたデータを MRU ポジションに移動するため，数パケットで構成されるフローのエントリがヒットした場合に当該エントリの追い出しが遅れる. ELC はヒット時のエントリ移動距離を 1 に制限する．ELC の概要を図6 (b) に示す．ELC では，例えば LRU ポジションから MRU ポジションに移動するためには，最短でも (way 数−1) 回のヒットを要する．これにより，パケット間隔の短いロングフローほどキャッシュに残ることが期待できる.

4.3 LIP および ELC の改良手法

更に，本報告ではこれら二つの手法の改善案を提案する．LIP と ELC は共に新規エントリを LRU ポジションに挿入するため，トラフィックの状況によってはロングフローであってもヒット前に置換される可能性がある．そこで，新規エントリの挿入位置を LRU ポジションから 1 だけ MRU 方向に変更した方式を LIP1, ELC1 として, LRU ポジションから 2 だけ MRU 方向に変更した方式を LIP2，ELC2 として提案する. 図6 (c) には，例として ELC1 の概要を示している. 本改善案により新規エントリが置換されるまでの猶予が増し，ネットワークによっては LSC や ELC より性能が向上すると考えられる.

5. 評価

前節で検討したライン置き換え方式に関して, PPC に適用した際のミス削減効果を評価した．本測定には，C++で実装した PPC シミュレータと，ワークロードとして複数の実ネットワークトラフィックのトレースを用いた．シミュレータの処理概要と構成パラメータを図7および表1に，ネットワークトレースの詳細を表2に示した. 本シミュレータは 1GHz の動作を仮定し，サイクルベースの処理を行う．Pcap または Erf 形式のトレースファイルからパケットのタイムスタンプ値に合わせてパケットを読み出し，フロー情報の抽出，CRC 関数によるキャッシュインデックスの生成，パケット処理キャッシュの参照，ミス時の TCAM の参照等をシミュレートする．図中の Cache Miss Table (CMT) は，TCAM でパケットを処

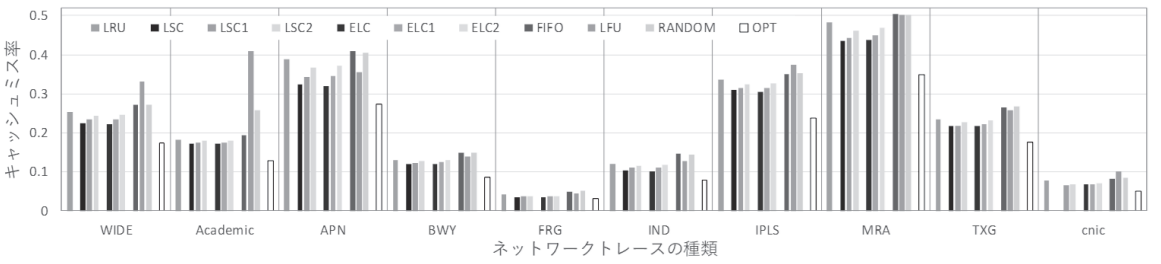

図8 シミュレーションによるキャッシュミス率の測定結果

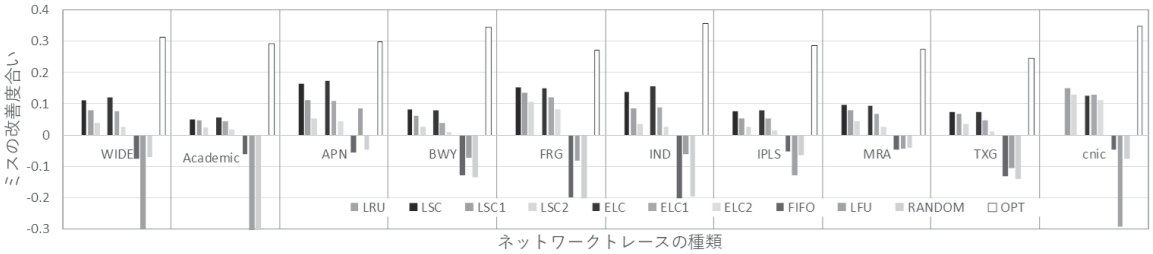

図9 LRUに対する各方式のキャッシュミスの改善度合い

表2 ネットワークトレースの詳細

トレース名	平均パケット数[pps] (packet per second)
WIDE[15]	24,657
Academic	371,013
APN[16]	20,793
BWY[16]	20,547
FRG[16]	32,722
IND[16]	15,540
IPLS[16]	99,264
MRA[16]	41,372
TXG[16]	11,610
cnic[16]	31,023

理中に同一フローの後続パケットが到着した場合に生じるミスを削減するため，後続パケットをCache Miss Queue（CMQ）に蓄える機構である．CMTおよびCMQの詳細に関しては論文[14]で述べている．評価に用いたワークロードはWidely Integrated Distributed Environment（WIDE）[15]およびRIPE Network Coordination Centre[16]より入手した．Academicトレースは国内のあるコアネットワークから採取した10Gbps回線のネットワークトレースである．

シミュレーションにより測定した各トレースの平均キャッシュミス率を図8に示した．また，図9はLRUに対する各方式のキャッシュミスの改善度合いを示している．図9より，全てのネットワークにおいてLSCとELCはLRUよりもミスを削減できており，特にELCを適用することでLRUに対し平均で11.1%，最大で17.4%のキャッシュミスを削減できることがわかった．一方で，FIFOやRandom方式は全てのトレースでミスが増大することから，パケット処理キャッシュに適さない．LFUはネットワークによってはLRU以上の性能が期待できるが，多くのトレースでミスが増大する．LSCとELCの改善案であるLSC1，LSC2，ELC1，ELC2は，通常のLSCやELCと比べ性能が低下した．これは，新規エントリの置換に猶予を与えることで増加するヒットよりも，ショートフローエントリが居残ることで増加するミスのほうが多いためであると考えられる．これらの結果より，パケット処理キャッシュにおいては，本報告で提案したELCを適用することで，最大の効果が得られることがわかった．また，OPTは全てのトレースでLRUに対し30%前後のキャッシュミスを削減可能であるという結果が得られている．これに対し，ELCの性能はOPTと比較し，平均で36.6%，最大で58.3%であり，ライン置き換え方式にはまだ改善の余地があることがわかった．

６．結論

ルータでは，TCAM へのアクセスが高い電力を消費し，なおかつパケット処理速度上のボトルネックとなっていた．これに対し，TCAM アーキテクチャの改善による省電力化アプローチは，TCAM の構造から大幅な省電力化が困難であった．そこで，キャッシュを用いて TCAM の使用頻度を削減することにより，ルータの省電力化，パケット処理高速化を達成するパケット処理キャッシュが提案されてきた．パケット処理キャッシュ機構は TCAM とのアクセス速度の兼ね合いから，プロセッサの L1 規模のキャッシュに実装することが望ましく，キャッシュミスが増大することが問題であった．

本研究では，パケット処理キャッシュにおいてキャッシュに格納するフローを適切に制御する種々の方法を提案し，キャッシュミスを削減した．その中でも，ライン置き換え方式に着目し，キャッシュ容量を増やすことなくキャッシュミスを削減した研究[5]について本報告では紹介した．本報告で提案した ELC を適用することで，一般的に用いられる LRU に対し，平均で 11.1%，最大で 17.4%のミスを削減できることを示した．これは，理想的な置き換え方式である OPT の 36.6%程度の性能であり，まだ ELC には改善の余地があることがわかった．

発表論文

[1] H. Yamaki, Flow-Length Aware Cache Replacement Policy for Packet Processing Cache, International Journal of Advanced Computer Science and Applications (IJACSA), Vol.9, No.5, pp.12-20, Jun. 2018.

[2] H. Yamaki, H. Nishi, S. Miwa, and H. Honda, Data Prediction for Response Flows in Packet Processing Cache, In Proceedings of 2018 55th ACM/EDAC/IEEE Design Automation Conference (DAC' 18), No.110, pp.1-6, Jun. 2018.

[3] H. Yamaki, Flow Characteristic-Aware Cache Replacement Policy for Packet Processing Cache, In Proceedings of Future of Information and Communication Conference (FICC 2018), pp.1-8, Apr. 2018.

[4] 八巻隼人，愛甲達也，三輪忍，本多弘樹，パケット処理キャッシュにおける送信元 IP アドレスに着目したミス削減手法に関する初期検討，情報処理学会研究報告 2017-ARC-226, Vol. 12, pp.1-8, 2017 年 5 月.

[5] H. Yamaki and H. Nishi, Line Replacement Algorithm for L1-scale Packet Processing Cache, In Adjunct Proceedings of 13th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS' 16), pp.12-17, Nov. 2016.

参考文献

（1） United Nations / Framework Convention on Climate Change, Adoption of the Paris Agreement, 21st Conference of the Parties, Paris: United Nations.

（2） “グリーン IT イニシアティブ － 経済産業省”. http://www.meti.go.jp/committee/materials/downloadfiles/g80520c03j.pdf, (参照 2019-02-28).

（3） Gamage, S. and Pasqual, A.. High performance parallel packet Classification architecture with Popular Rule Caching. 18th IEEE International Conference on Networks (ICON). 2012, pp.52-57.

（4） Agrawal, B. and Sherwood, T.. Ternary CAM power and delay model: extensions and uses. IEEE Trans. Very Large Scale Integr. Syst.. 2008, vol.16, no.5, pp.554-564.

（5） Nawa, M. et al.. Energy-efficient high-speed search engine using a multi-dimensional TCAM architecture with parallel pipelined subdivided structure. 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). 2016, pp.309-314.

（6） “Energy Efficient Networking - Business white paper”, http://h17007.www1.hp.com/docs/mark/4AA3-3866ENW.pdf (参照 2019-02-28).

（7） Mishra, T. B. and Sahni, S.. PETCAM-a power efficient tcam architecture for forwarding tables. IEEE Trans. Computers. 2012, vol.61, no.1, pp.3-17.

（8） Pagiamtzis, K. and Sheikholeslami, A.. Content-addressable memory (CAM) circuits and architectures: a tutorial and survey. in IEEE Journal of Solid-State Circuits. 2006, vol.41, no.3, pp.712-727.

（9） Yamaki, H. and Nishi H.. An Improved Cache Mechanism for a Cache-based Network Processor. Journal of Communication and Computer. 2013, vol.10, no.3, pp.277-286.

（１０） Talbot, B. et al.. IP caching for terabit speed routers. Global Telecommunications Conference (GLOBECOM). 1999, vol.2, pp.1565-1569.

（１１） “Intel 64 and IA-32 Architectures Optimization Reference Manual September 2015”. http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-optimization-manual.pdf, (参照 2019-02-28).

（１２） Belady, L. A.. A study of replacement algorithms for a virtual-storage computer. IBM Syst. J.. 1966.

（１３） Qureshi, M. K. et al.. Adaptive insertion policies for high performance caching. Proc. 34th annual international symposium on Computer architecture (ISCA). 2007, pp.381-391.

（１４） Okuno, M. and Nishi, H.. Network-Processor Acceleration-Architecture Using Header-Learning Cache

and Cache-Miss Handler. The 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI). 2004, vol.3, pp.108-113.
（１５）“MAWI Working Group Traffic Archive – WIDE MAWI WorkingGroup”. http://mawi.wide.ad.jp/mawi/ (参照 2019-02-28).
（１６）“RIPE Network Coordination Centre”. http://www.ripe.net/ (参照 2019-02-28).